NAME: **login ID:**
precept:

COS 126 Written Exam 2, Fall 2010

This test is 9 questions, weighted as indicated. The exam is closed book, except that you are allowed to use a one page cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. *Put your name, login ID, and precept number on this page* (*now*), and write out and sign the Honor Code pledge before turning in the test. You have 50 minutes to complete the test.

"I pledge my honor that I have not violated the Honor Code during this examination."

Signature

1	Binary search trees	/8
2	REs/DFAs	/10
3	Linked Structures	/10
4	ADTs and APIs	/10
5	Divide and conquer	/8
6	Gates	/5
7	Components	/5
8	Theory	/16
9	Turing machines	/8
TOTAL		/80

December 16, 2010

1. **Binary Search Trees** (8 points).

   Suppose that you are searching for the key `70` in a binary search tree. In the following list, circle the letters corresponding to sequences that could *not* be the sequence of keys examined.

   A. 70

   B. 99 10 80 20 60 70

   C. **10 80 40 32 50 70**

   D. 10 26 30 48 50 62 70

   E. **41 99 20 85 70**

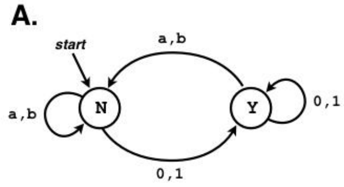
   F. 44 80 55 61 70

   G. 99 11 53 86 70

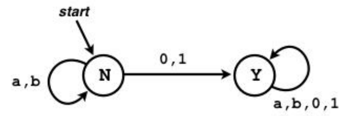
   H. 22 58 81 70

2. **REEs/DFAs** (10 points). For this problem the alphabet consists of the two letters a and b and the two numbers 0 and 1. Consider the following DFAs (identified by letters) and REs (identified by numbers).

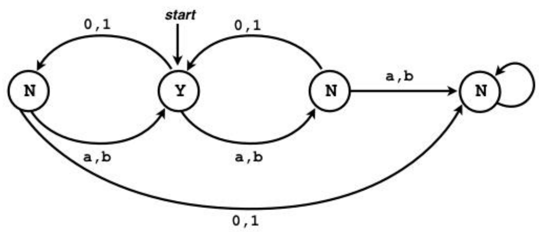
A.



B.



C.




```
1.    (a|b)(a|b)*(0|1)(a|b|0|1)*(a|b)

2.    (a|b)(a|b)*(0|1)(a|b)*(a|b)

3.    (a|b)*(0|1)(a|b|0|1)*

4.    ((a|b)(0|1))* | ((0|1)(a|b))*
```

For each of the descriptions below write a letter in the first column if the corresponding DFA recognizes all the strings in the given set (and only those strings!) and write a number in the second column if the corresponding RE is an equivalent description of the set. If no RE or DFA does the job write "none" (blanks are considered to be incorrect).

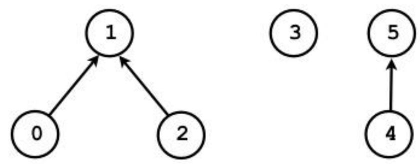
DFA	RE	
__B__	__3__	All strings with at least one number
none	__4__	All strings of even length in which letters and numbers alternate
__A__	none	All strings that end in a number
none	__1__	All strings that begin and end with letters and have at least one number

3. **Linked structures** (10 points). Consider the following Java class, which implements a linked data structure, and then answer the questions on the next page.

```
public class Forest
{
   private Node[] links;

   private class Node
   {
      private Node next;
   }

   public Forest(int N)
   {
      links = new Node[N];
      for (int i = 0; i < N; i++)
         links[i] = new Node();
   }

   private Node root(int i)
   {
      Node x = links[i];
      while (x.next != null) x = x.next;
      return x;
   }

   public void merge(int i, int j)
   {
      root(i).next = root(j);
   }

   public boolean merged(int i, int j)
   {
      return root(i) == root(j);
   }
}
```

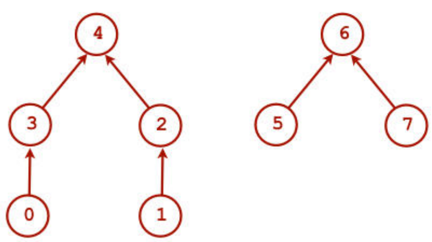
3 (continued). `Forest` builds general structures known as *forests*, where roots have null links and every other node points to its parent. To begin to understand how it works, verify that the client code at left produces the forest drawn below (each node is labeled with the index in the `links` array that has a reference to it).

```
Forest t = new Forest(6);
t.merge(0, 1);
t.merge(2, 1);
t.merge(4, 5);
```



A. (6 points) Draw the forest that is created by the following client code:

```
Forest t = new Forest(8);
t.merge(0, 3);
t.merge(1, 2);
t.merge(1, 4);
t.merge(5, 6);
t.merge(3, 4);
t.merge(7, 5);
```



B. (4 points) Now suppose that the following calls to `t.merged()` follow the client code of part A. In the blank to the left of each call, write `true` or `false` to give the value returned.

__ **true**__ `t.merged(0, 3);`

__**false**__ `t.merged(0, 7);`

__ **true**__ `t.merged(1, 3);`

__**false**__ `t.merged(4, 5);`

4. **ADTs and APIs** (10 points).

A. (3 points) Give an API for the `Forest` class in question 3. For full credit, include comments.

```
        Forest(int N)      Create a forest of items 0 to N-1.
   void merge(int i, int j)    Put i and j into the same tree.
boolean merged(int i, int j)   Are i and j in the same tree?
```

B. (7 points) In the blanks to the left of each of the descriptions at the bottom of the page write the letters corresponding to the identifiers from `Forest` that match the description. In your answer, some blanks may be empty and others may have multiple letters.

A. `links`

B. `Node`

C. `Forest`

D. `root`

E. `merge`

F. `x`

__**E C**__ Any client can refer to it.

__ **A** __ Instance variable.

_ **D E** _ Instance method.

F D B A No client can refer to it.

__**B C**__ Data type.

__**A F**__ Reference variable.

__ **B** __ Inner class.

5. **Divide and conquer** (8 points). Consider the following class:

```
public class Mystery
{
   public static String twist(String s)
   {
      if (s.length() < 2) return s;
      int m = s.length()/2;
      String lh = twist(s.substring(0, m));
      String rh = twist(s.substring(m, s.length()));
      s = rh + lh;
      System.out.println(s);
      return s;
   }

   public static void main(String[] args)
   {
      String s = args[0];
      s = twist(s);
   }
}
```

Recall that `s.substring(i, j)` returns the substring of `s` from indices `i` to `j-1`. For example, if `s` is the string `"stressed"` then `s.substring(0, 6)` is `"stress"`.

A. Give the one line of output produced by the command

```
% java Mystery ab
```

___**ba**___

B. Give the seven lines of output produced by the command

```
% java Mystery stressed
```

___**ts**___

___**er**___

__**erts**__

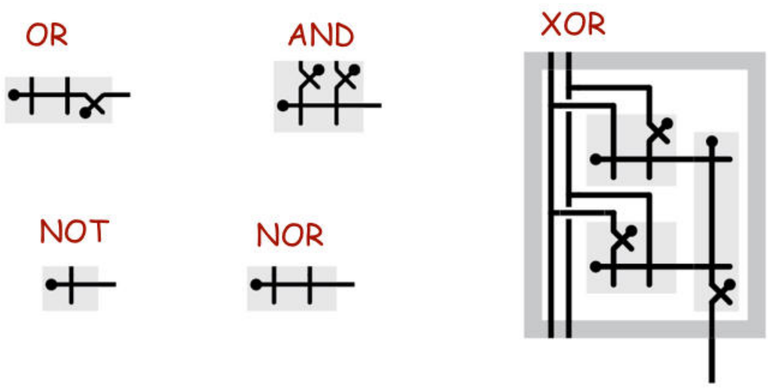
___**ss**___

___**de**___

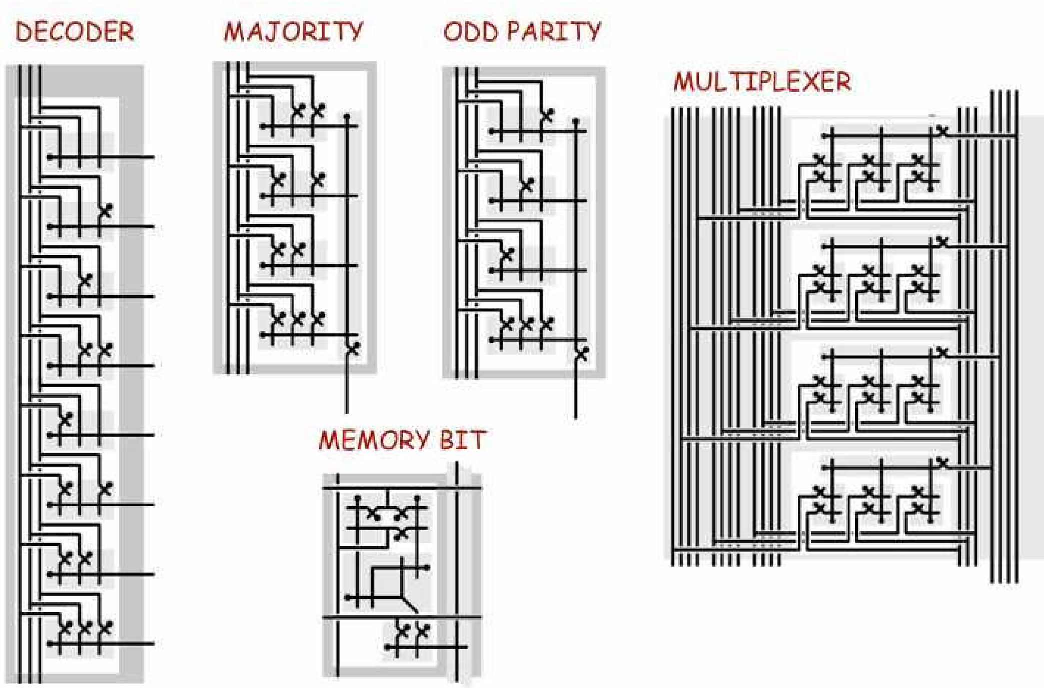
__**dess**__

desserts

6. **Gates** (5 points). Identify the gates below by writing AND, OR, NOT, NOR, or XOR above each of them.



7. **Components** (5 points). Identify the circuit components below by writing DECODER, MAJORITY, MULTIPLEXER, MEMORY BIT, or ODD PARITY above each of them.



8. **Theory** (16 points). Match the theoretical results, conjectures, theses, and ideas (A-H) with their practical consequences, by writing a letter in the blank to the left of each. Assume that "fast" means "in polynomial time," "TSP" stands for the traveling salesperson problem, and "SAT" stands for the boolean satisfiability problem. In this problem, each letter should be used once and only once.

A. Cook's Theorem
B. Turing's Entscheidungsproblem (decision problem) paper, 1937
C. The Church-Turing thesis
D. P = NP
E. There is a DFA that accepts the same language as any given RE
F. Karp's reductions in his landmark 1972 paper
G. Factoring is hard
H. P is not equal to NP

__C__ My laptop can solve any problem that yours can.

__F__ If you can solve the TSP fast, then you can also solve SAT fast.

__D__ SAT can be solved fast.

__G__ Online banks can retain their current encryption programs.

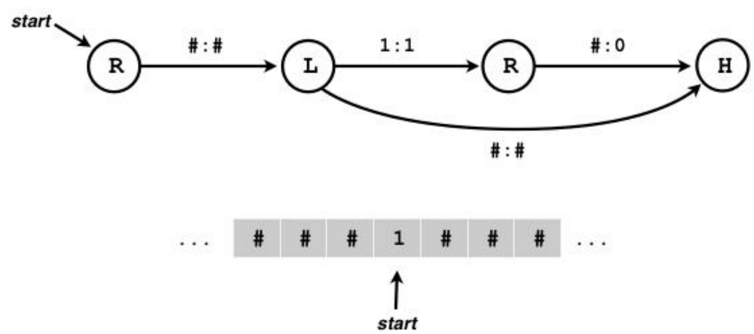
__H__ The TSP cannot be solved fast.

__B__ Some problems cannot be solved by following a set of mechanical instructions.

__A__ If you can solve SAT fast, then you can also solve the TSP fast.

__E__ It is easy to write a program that searches for email addresses on a web page.

9. **Turing machine (8** points). Consider the following Turing machine.



A. (4 points) Give the contents of the tape after the machine halts when it is run for the given input tapes, with the head initially positioned at the leftmost `1`. (*Reminder*: the tape head moves on *entering* a state.)

```
input:    ...# # # 1 # # # ...
```

```
result: __ ...# # # 1 0 # # # ... ___
```

```
input:    ... # # # 1 0 1 # # # ...
```

```
result: __...# # # 1 0 1 0 # # # ...___
```

B. (2 points) What function does this machine compute (as a mathematical function of x, where x is the binary number on the tape)?

2x

C. (2 points) Suppose that your computer is not connected to any external (wireless or wired) source. What can this Turing machine do that your computer cannot do?

Compute 2x for arbitrarily large x.